UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/494,876	01/31/2000	Michael A. Crane	AUS000085US1	4071

35525 7590 05/20/2004

DUKE W. YEE
CARSTENS, YEE & CAHOON, L.L.P.
P.O. BOX 802334
DALLAS, TX 75380

EXAMINER	
SIMITOSKI, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2134	5

DATE MAILED: 05/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/494,876	Applicant(s) CRANE ET AL.	
	Examiner Michael J Simitoski	Art Unit 2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 February 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,5-17 and 21-26* is/are rejected.

7)☒ Claim(s) *2-4 and 18-20* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

NORMAN M. WRIGHT
PRIMARY EXAMINER

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. The amendment of 2/10/04 (paper #4) has been received and considered.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim 13 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claim is indefinite because it is unclear whether "the keystore is within a plurality of keystores" means that the a keystore is encapsulated within a plurality of other keystores, such that the plurality of keystores each contain a copy of the keystore or the keystore is simply one of a plurality of keystores in the system. As best understood in light of paper #4, *for the purposes of this Office Action, the limitation is understood to mean that the keystore is one of a plurality of keystores.*

Claim Rejections - 35 USC § 103

4. Claim 1, 5-7, 17, 21-23 & 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Release 1.0 of Intel's Common Data Security Architecture, partly described in "Intel's Common Data Security Architecture", by Intel Corporation (Intel) published December 1996 in view of U.S. Patent 6,625,734 to Marvit et al. (Marvit).

Regarding claims 1, 5, 17, 21 & 26, Intel discloses receiving a call from an application to perform an operation (pages 10, 14 & 20). While Intel does not specifically disclose using a key

in the functions (pages 10-11), the examiner takes Official Notice that functions such as CL_CertSign() (page 18) and many methods in the Java Security classes (page 34) using keys is old and well established in the art of security as a method of performing cryptographic transformations on data. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a function that requires a key. One of ordinary skill in the art would have been motivated to perform such a modification to perform cryptographic transformations on data. This advantage is well known to those skilled in the art. Intel discloses in response to receiving the call from the application (page 10), automatically identifying/selectively dispatching a routine/function(s) to perform the operation (through application programming interface) (pages 10-11) and creating a data structure/security context (page 14) used by the routine/function(s) (page 14) to execute the operation (page 14), wherein the data structure/context includes parameters/stateful information (page 13) of the call received from the application (page 14) and sending the data structure to the routine (page 14). Intel lacks automatically identifying a keystore containing the key. However, Marvit teaches that in a system of multiple users on a network, there is a need to control usage and track the information disseminated (col. 2, lines 1-5), which can be accomplished by storing keys needed for encryption and decryption at a key repository and issuing the keys as needed to requestors (Fig. 1). In a system with multiple key repositories, the requestor requests a message and receives the repository/keystore ID (automatically identifying the repository/keystore). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to automatically identify the keystore containing the key. One of ordinary skill in the art would

have been motivated to perform such a modification to control the dissemination of data on a network, as taught by Marvit (col. 2, lines 1-5, col. 12, lines 39-54 & Figs. 1 & 5a).

Regarding claims 6 & 22, Intel discloses a common data security architecture plug-in (page 12).

Regarding claims 7 & 23, Intel does not explicitly disclose initializing the routine/function(s) prior to sending the data structure/context, the examiner takes Official Notice that initializing a function (preparing memory structures, etc.) before passing data to the function is old and well established in the art of application programming. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to initialize the routine before sending the data structure. One of ordinary skill in the art would have been motivated to perform such a modification because it is well established to do so by those skilled in the art.

5. Claims 8 & 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Intel in view of Marvit, as applied to claims 1 & 17 above, in further view of "Common Security Services Manager: Service Provider Interface Specification" by Intel (Intel SPI). Intel, as modified above, lacks explicitly receiving a result from the operation and returning it to the application. However, Intel SPI teaches that CSPs (with associated routines) perform cryptographic services (§3.1) and return the results to the applications (§3.4.2). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to receive the results from the operation and return the results to the application. One of

ordinary skill in the art would have been motivated to perform such a modification perform cryptographic operations in the CDSA system, as taught by Intel SPI (§3.1 & §3.4.2).

6. Claims 9 & 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Intel in view of Marvit and Intel SPI, as applied to claims 8 & 24 above, in further view of Sun Microsystems's JAVA Platform v1.2 (JDK 1.2), published December 1999, partly described in "Java Platform 1.2 API Specification: Class KeyStore" (Sun). Intel, as modified above, discloses that the Java security package can be used with CDSA and the CSSM (pages 32-34). Intel, as modified above, lacks specifically updating the keystore. However, Sun teaches that the Java API supports keystores to hold collections of keys and certificates (page 1) and supports methods such as *deleteEntry()* (page 2) to update the keystore. Further, Marvit teaches a system as described above, further teaching that it is useful to render the messages inaccessible by deleting keys from the repository (col. 6, lines 51-61) (after a last validated user has received the key). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to perform updates to the keystore in response to receiving the result from the routine. One of ordinary skill in the art would have been motivated to perform such a modification to use the Java security package, as suggested by Intel (pages 32-34) to manage collections of keys and certificates, as taught by Sun (pages 1-2) to render the use of data requiring the keys inaccessible if desired, as taught by Marvit (col. 6, lines 41-51).

7. Claims 10-12, 14 & 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Intel in view of Sun Microsystems's JAVA Platform v1.2 (JDK 1.2), partly described in "Java

Platform 1.2 API Specification: Class KeyStore" (API Spec) and "VM Spec Structure of the Java Virtual Machine" (VM Spec) (Sun).

Regarding claims 10 and 11, Intel discloses a security layer/CSSM and a plurality of cryptographic routines/CSPs accessed through a security layer/CSSM (see page 33). Intel further discloses applications calling routines/services through the security layer/CSSM to perform cryptographic operations and receiving the results (see page 10) where the CSSM Security API automatically identifies routines/functions (page 10), but lacks a keystore and a keystore application program interface layer coupled to the security layer. However, Intel discloses a "Java to CSSM Wrapper"/keystore application program interface layer (see page 33). Referring to the API Spec, Sun discloses a keystore class for creating "keystores", enabling the storage and management of cryptographic keys (see page 1). Referring to VM Spec, Sun teaches that JDK 1.2 offers the benefits of multi-platform compatibility, small code size and user security (see § Introduction, paragraph 7 titled "The Java Virtual Machine"). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include a keystore and "Java to CSSM Wrapper"/keystore application program interface layer in the CDSA to manage keys. One of ordinary skill in the art would have been motivated to perform such a modification, as suggested by Intel, to gain the benefits of Java's multi-platform compatibility, small code size and user security, as taught by Sun.

Regarding claim 12, Intel discloses a plurality of plug-ins/CSP modules (see page 10).

Regarding claim 14, Sun discloses methods that operate upon a particular keystore through the Java API, which calls underlying routines to perform the instructions of the methods (see pages 2-3 of API Spec).

Regarding claim 16, Sun discloses that when a Java method is instantiated, a Java Virtual Machine frame/data structure is created to store both data and partial results used by the method/routine (see § 3.6, VM Spec).

8. Claims 13 & 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Intel in view of Sun, as applied to claim 10 above, in further view of Marvit.

Regarding claim 13, Intel, as modified above, lacks the keystore being one of a plurality of keystores and the keystore application program interface layer identifying the keystore from a plurality of keystores. However, Marvit teaches that in a system of multiple users on a network, there is a need to control usage and track the information disseminated (col. 2, lines 1-5), which can be accomplished by storing keys needed for encryption and decryption at a key repository and issuing the keys as needed to requestors (Fig. 1). In a system with multiple key repositories/keystores, the requestor requests a message/result and receives the repository/keystore ID (identifying the repository/keystore). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to combine the inventions of Intel, as modified above, and Marvit to identify the keystore from a plurality of keystores. One of ordinary skill in the art would have been motivated to perform such a modification to control the dissemination of data on a network, as taught by Marvit (col. 2, lines 1-5, col. 12, lines 39-54 & Figs. 1 & 5a).

Regarding claim 15, Sun discloses methods such as *setKetEntry* and *deleteEntry* for the purposes of updating the keystore (see pages 7 and 8, API Spec). Sun lacks updating the keystore in response to receiving the result from the routine. However, Marvit teaches that it is

useful to render the messages inaccessible by deleting keys from the repository (col. 6, lines 51-61) (after a validated user has received the key/receiving results). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to perform updates to the keystore in response to receiving the result from the routine. One of ordinary skill in the art would have been motivated to perform such a modification to use the Java security package, as suggested by Intel (pages 32-34) to manage collections of keys and certificates, as taught by Sun (pages 1-2) to render the use of data requiring the keys inaccessible if desired, as taught by Marvit (col. 6, lines 41-51).

Response to Arguments

9. Applicant's arguments with respect to claims 1-26 have been considered but are moot in view of the new ground(s) of rejection.

10. In light of applicant's amendment of 2/10/04, the objections to claims 3, 13 & 19 are withdrawn.

11. The proposed drawing corrections of 2/10/04 have been accepted and entered and therefore the objection to the drawings is withdrawn.

12. Regarding applicant's arguments on page 16, with respect to the listed security classes, while Intel does not specifically disclose use of the keystore class, the keystore class was soon thereafter incorporated into the Java security package to which the Intel reference is referring. The Intel reference, when used with Java, performs the identification of a routine on page 10 and calling the identified routine to perform the cryptographic functions on pages 10 and 11. As seen

on page 33, applications make calls, which are received at the Java-CSSM wrapper and the correct CSSM routines (in the add-in security modules) are identified.

13. The examiner also brings the following to applicant's attention (regarding the independent claims1, 17 & 26):

If a Java Keystore object is invoked (which takes a KeyStoreSpi object as an argument) (page 2, Class KeyStore), a KeyStoreSpi class method call (such as *KeyStoreSpi.engineAliases()*) would automatically identify the keystore in response to the call (page 1, Class KeyStoreSpi). Similarly, a KeyStoreSpi class method call such as *KeyStoreSpi.engineDeleteEntry()* would automatically identify the keystore and perform updates to said keystore in response to receiving a call. While an application implementation of this method is not explicitly shown in the reference, the Java security package is shown in the Intel reference to be a beneficial addition to the CDSA and therefore a skilled artisan would have been able to use any function in the Java security package according to its intended use.

Allowable Subject Matter

14. Claims 2-4 & 18-20 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. The Sun references were cited for teaching implementations of the Java Security API and Java Cryptography Architecture.

b. The Wood reference was cited for teaching an adaptation layer for PKCS #11 (a cryptographic service provider).

c. The '264 & '736 patent references were cited for teaching distributed key management where keys are distributed and the locations of the keys need to be identified.

16. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Simitoski whose telephone number is (703)305-8191.

The examiner can normally be reached on Monday - Thursday, 6:45 a.m. - 4:15 p.m.. The examiner can also be reached on alternate Fridays from 6:45 a.m. – 3:15 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached on (703)308-4789.

Any response to this action should be mailed to:
Commissioner of Patents and Trademarks
Washington, DC 20231
Or faxed to:
(703)746-7239 (for formal communications intended for entry)
Or:
(703)746-7240 (for informal or draft communications, please label "PROPOSED" or "DRAFT")
Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA 22202, Fourth Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MJS
April 8, 2004

NORMAN M. WRIGHT
PRIMARY EXAMINER